

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-306007

(43)Date of publication of application : 05.11.1999

(51)Int.Cl.

G06F 9/06
G06F 13/10

(21)Application number : 10-131122

(71)Applicant : NEC CORP

(22)Date of filing : 24.04.1998

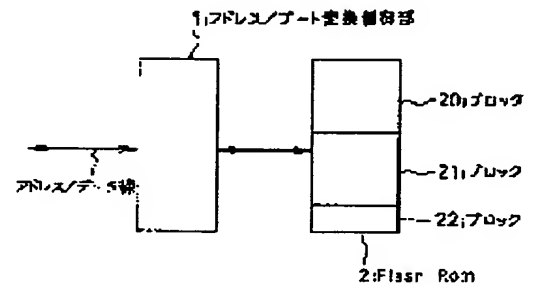
(72)Inventor : MORIMITSU HIROSHI

(54) METHOD AND SYSTEM FOR BIOS REWRITING

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a basic input output system(BIOS) rewriting method and its system which permit activation of a system from the next time onward even if rewriting of a BIOS data fails.

SOLUTION: A flash ROM is divided into blocks 20 and 21 for storing a BIOS and a block 22 for storing status information, a block to read the BIOS on the basis of information first read from the block 22 at the time of activation is selected and the system is started, and a block different from ones used for starting the system out of the blocks 20 and 21 that store the BIOS is rewritten by a new BIOS at the time of BIOS rewriting. Then, the device is equipped with an address/boot conversion control means (1) which controls so that the BIOS is read out of the block in which the BIOS is rewritten anew at the time of next system start and the system is started at the next time of starting the system only when rewriting is normally performed by the new BIOS.



LEGAL STATUS

[Date of request for examination] 24.04.1998

[Date of sending the examiner's decision of rejection] 26.09.2000

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

METHOD AND SYSTEM FOR BIOS REWRITING

5 [Claims]

[Claim 1] A BIOS rewriting method, comprising steps of:

dividing a re-writable non-volatile semiconductor storage device into a plurality of areas for storing a BIOS respectively and an area which stores status information including information for determining an area to be read for system startup and is read first at the time of system startup;

starting up the system by first selecting an area to read the BIOS based on the information read from the area for storing said status information and reading out the BIOS from said selected area at the time of system startup;

rewriting an area, which is different from said area used for system startup out of the plurality of areas storing said BIOS, by a new BIOS at the time of BIOS rewriting; and

20 reading the BIOS from said area for which the BIOS was newly rewritten and starting up the system at the next system startup only when rewriting is normally performed by the new BIOS.

[Claim 2] A BIOS rewriting method, comprising steps of:

25 dividing a flash ROM into a block for storing status information which is read first at a system startup, and a plurality of blocks for storing a BIOS;

starting up the system by selecting a block of which the BIOS is read from the information, which is read from the block storing the status information, and reading said BIOS from the selected block at the time of system startup;

5 rewriting a block which is different from said block used for system startup (called the "original block"), by a new BIOS at the time of BIOS writing;

 updating the content of the block for storing said status information when rewriting is normally performed by
10 said new BIOS; and

 reading the BIOS from said block for which the BIOS was newly rewritten by the new BIOS and starting up the system at the next startup when rewriting of said BIOS was normally performed, or reading the BIOS from said original block and
15 starting up the system at the next startup when rewriting of said BIOS failed.

[Claim 3] A BIOS rewriting system, comprising address/boot conversion control means for:

 dividing a rewritable non-volatile semiconductor
20 storage device into a first and second areas for storing a BIOS and a third area for storing status information which is read first at a system startup; and

 performing control such that the system is started up by reading first the status information stored in said
25 third area in said non-volatile semiconductor storage device to control the address, and reading BIOS data written in either said first or second area at the time of the system

startup, and that the new BIOS is written to an area which is different from the area used for system startup out of said first and second areas in said non-volatile semiconductor storage device at the time of BIOS rewriting, and updating said third area by a latest status information on which one of said first and second areas the system is started up from.

[Claim 4] The BIOS rewriting system according to Claim 3, wherein said address/boot conversion control means further comprises:

status information storing means which includes a boot flag for indicating which BIOS stored in said first or second area is read for starting up the system and an update flag for indicating the BIOS rewriting mode;

means of reading first the status information from said third area in said non-volatile semiconductor storage device and storing the content thereof in said status information storage means at the time of system startup, and inverting the value of said boot flag and writing the value of the status information storage means to said third area as

a new status information after rewriting of the BIOS; and

means of converting an address so that either said first or second area is accessed according to the value of said boot flag if said update flag is not rewriting mode;

and converting an address such that an area opposite of the area used for reading the BIOS at the system startup out of said first and second areas is accessed if the value of said update flag indicates rewriting mode.

[Claim 5] The BIOS rewriting system according to Claim 3, further comprising means for switching and setting the value of said boot flag to a fixed value outside said address/boot conversion control means.

5 [Claim 6] A BIOS rewriting system, comprising address/boot conversion means for:

dividing a re-writable non-volatile semiconductor storage device into N number of areas (N is a 3 or higher integer) for storing a BIOS, and N+1 number of areas for

10 storing said status information which is read first at a system startup;

starting up the system by reading BIOS data written in one of said first to Nth areas, with controlling the address, from the status information stored in the
15 (N+1)th area in said non-volatile semiconductor storage device;

performing control such that the new BIOS is written to an area which is different from the area used for the system startup out of the first to Nth areas in said non-volatile semiconductor memory device when the BIOS is
20 rewritten, and said (N+1)th area is updated to the latest status on which one of the first to Nth areas the system is started up from.

[Claim 7] The BIOS rewriting system according to one of
25 Claims 3 to 5, wherein said rewritable non-volatile semiconductor storage device further comprises a flash ROM which can electrically erase and write for each block.

[Detailed Description of the Invention]

[0001]

[Field of the Invention]

The present invention relates to a BIOS (Basic
5 Input/Output System) rewriting system of such an information
processor as a personal computer, and more particularly to a
BIOS rewriting system which is suitable for a system storing
BIOS in an erasable and writable flash ROM in block units.

[0002]

10 [Prior Art]

When a BIOS, which is basic software to control hardware
functions of such an information processor as a personal
computer, is stored in a mask ROM, for example, update
thereof is performed by replacing the mask ROM, but recently
15 an information processor, where BIOS is stored in a flash ROM
which is rewritable in block units, and where a BIOS can be
updated for correction and function additions, has been
commercialized. In this case, to rewrite BIOS, the user
themselves updates it or a person-in-charge of maintenance
20 performs this update if the maintenance contract has been
signed.

[0003]

[Problems that the Invention is to Solve]

However in order to rewrite BIOS, it is necessary to
25 obtain BIOS, startup a predetermined rewriting tool, and
perform rewriting according to a predetermined procedure, and

rewriting may fail unless there exists a certain degree of specialized knowledge.

[0004]

Also the system cannot be started up normally at the
5 next startup if power is shut off in error during rewriting
or if BIOS is not written normally to a flash ROM after the
BIOS data in the flash ROM is erased by a rewriting tool.
This is because a system is started up by first reading the
BIOS data in the flash ROM, but a system cannot be started up
10 if the BIOS data is lost or destroyed.

[0005]

As a BIOS rewriting system, Japanese Patent Application
Laid-Open No. H8-69376 proposes a BIOS rewriting device for
preventing re-operation from being disabled even if power is
15 shut off when rewriting the content of the BIOS ROM,
comprising an address switching circuit, which switches
between the higher and lower parts of a non-volatile memory
device storing the BIOS and which has the same capacities, is
supported using a jumper switch, wherein the BIOS is divided
20 into a basic part and an extension part, the basic part is
rewritten by writing the new basic part to the part storing
an old extension part, replacing the higher and lower parts
of the non-volatile memory device as circuits after writing
is over, then the block storing the old BIOS basic part is
25 erased and a new extension part is written. As clarified in
the later mentioned explanation, the present invention
presents a system which is different from the rewriting

system stated in the above mentioned Japanese Patent
Application Laid-Open H8-69376.

[0006]

With the foregoing in view, it is an object of the
5 present invention to provide a method and a system for BIOS
rewriting, which allows starting up the system the next time
or later even if rewriting the BIOS data failed.

[0007]

[Means of Solving the Problems]

10 To achieve the above mentioned object, a BIOS rewriting
method according to the present invention comprises steps of:
dividing a rewritable non-volatile semiconductor storage
device into a plurality of areas for storing a BIOS
respectively and an area which stores status information
15 including information for determining an area to be read for
system startup and is read first at the time of system
startup; starting up the system by first selecting a block to
read the BIOS based on the information read from the area for
storing the status information and reading out the BIOS from
20 the selected block at the time of system startup; rewriting
an area, which is different from the area used for system
startup out of the plurality of areas storing the BIOS, by a
new BIOS at the time of BIOS rewriting; and reading the BIOS
from the area for which the BIOS was newly rewritten and
25 starting up the system at the next system startup only when
rewriting is normally performed by the new BIOS.

[0008]

A BIOS rewriting system according to the present invention comprises address/boot conversion control means for dividing a rewritable non-volatile semiconductor storage device into a first and second areas for storing a BIOS and a third area for storing status information which is read first at a system startup, and performing control such that the system is started up by reading first the status information stored in the third area in the non-volatile semiconductor storage device to control the address, and reading BIOS data written in either the first or second area at the time of the system startup, and that the new BIOS is written to an area which is different from the area used for system startup out of the first and second areas in the non-volatile semiconductor storage device at the time of BIOS rewriting, and the third area is updated by a latest status information on which one of the first and second areas the system is started up from.

[0009]

[Description of the Preferred Embodiments]

Embodiments of the present invention will now be described. In the present invention, the flash ROM is divided into a block for starting up the system and a block for rewriting a BIOS, and only when rewriting is normally performed, the system is started up from the block where the BIOS was newly rewritten.

[0010]

Describing this in more detail, in the BIOS rewriting method of the present invention according to this preferred embodiment thereof, the flash ROM for storing a BIOS is divided into a block for storing status information which is read first at a system startup, and a plurality of blocks for storing BIOS, the system is started up by selecting a block for reading out the BIOS from the information read from the block storing the status information, and reading the BIOS from the selected block at the time of system startup, a block which is different from the block used for system startup (called "original block") is rewritten by a new BIOS at the time of BIOS rewriting, the content of the block for storing the status information is updated when rewriting is normally performed by the new BIOS, and the BIOS is read from the block for which the BIOS was newly rewritten by the new BIOS, to startup the system at the next startup if rewriting of the BIOS was normally performed, or the BIOS is read from the original block to startup the system at the next startup when rewriting of the BIOS failed.

[0011]

Referring to Fig. 1, the BIOS rewriting system of the present invention according to the preferred embodiment thereof comprises address/boot conversion control means (1) for dividing a flash ROM (2) into a first and second blocks (20, 21) for storing a BIOS, and a third block (22) for storing status information which is read first at a system startup, and performing control such that the system is

started up by reading first the status information stored in the third block and reading BIOS data written in either the first or second block by controlling the address at the time of system startup, and that the new BIOS is written to an area which is different from the area used for system startup out of the first and second blocks in the flash ROM (for example, if the system was started up from the first block, BIOS is rewritten to the second block) at the time of BIOS rewriting, and the third block is updated by the latest status information on which one of the first and second blocks the system is started up from.

[0012]

In other words, the address/boot conversion control means (1) further comprises functional means for controlling the address from the status information stored in the block (22) in the flash ROM (2), controlling such that the system is started from the BIOS data in either block (20) or block (21).

[0013]

Also at the rewriting of the BIOS, the address/boot conversion control section (1) further comprises the function means for controlling such that [the data] is written to a block which is different from the block used for system startup, and controlling such that the block (22) is updated to the latest information.

[0014]

According to an embodiment of the present invention, the flash ROM comprises a plurality of blocks storing BIOS data, and at the rewriting of the BIOS, the BIOS in a block different from the block used for the system startup is rewritten, and the block used for the system startup is not erased or written, so even if the rewriting of the BIOS failed, generation of a situation where the system cannot be started up at the next system startup can be prevented with certainty. This will now be described using examples.

[0015]

[Embodiments]

Embodiments of the present invention will now be described with reference to the accompanying drawings. Fig. 1 is a diagram depicting the configuration of an embodiment of the present invention.

[0016]

Referring to Fig. 1, an embodiment of the present invention comprises an address/boot conversion control section 1 for converting an address and a block to be read in the flash ROM 2 between the address/data line and the control line which are connected to the flash ROM 2 (batch erase type read only memory which can be electrically erased and written), where BIOS data is written and the stored content is held even if power is shut off.

[0017]

The flash ROM 2 is comprised of two blocks, 20 and 21, where BIOS data is written and the data can be read and

written, and the block 22, in which the status information of the flash ROM 2 is stored and which is read first at system startup.

[0018]

5 Fig. 2 is a block diagram depicting an example of the configuration of the address/boot conversion control section 1 according to an embodiment of the present invention.

Referring to Fig. 2, the detailed configuration of the address/boot conversion control section 1 according to an
10 embodiment of the present invention will be described.

[0019]

The address/boot conversion control section is comprised of a status read/write section 10 which is connected to the address/data line and the control line, for controlling the
15 reading/writing of status information from the block 22 of the flash ROM 2, a status register 12 which can store the values which are read from the status read/write section 10, and an address conversion section 11 which read and judges the status register 12 and converts the address.

20 [0020]

The status read/write section 10 has a function to read the status information first from the block 22 and save it in the status register 12 when the system is started up from the BIOS data in the block 20 or block 21, and to write the set
25 value of the status register 12 in the block 22 as new status information after rewriting completes.

[0021]

The status register 12 comprises a boot flag 120 which is set to "0" when the system starts up with the BIOS data of the block 20, and to "1" when the BIOS data is of the block 21, and an update flag 121 which is set to "1" in the case of
5 rewrite mode, and to "0" when otherwise.

[0022]

The address conversion section 11 has a function to convert the address so as to (1) access the block 20 when the value of the update flag 121 is "0" and the value of the boot
10 flag 120 is "0", and to (2) access the block 21 when the value of the update flag 121 is "0" and the value of the boot flag 120 is "1", and to convert an address to access a block opposite of the block used for the update when the value of the update flag 121 is "1".

15 [0023]

Fig. 3 is a flow chart depicting the operation of an embodiment of the present invention. The operation of the embodiment of the present invention will be described with reference to Fig. 1 to Fig. 3.

20 [0024]

When the power of the system is turned on (step S1), the status read/write section 10 reads the status information from the block 22, and stores the information in the status register 12 (step S2). The address conversion section 11
25 starts up the system by reading the block 20 if the value of the boot flag 120 is "0" and the block 21 if "1".

[0025]

The program starts up from a boot medium, such as a boot FD (Floppy Disk) or boot disk where data of a new BIOS and rewriting tool are stored, and rewriting starts.

[0026]

5 When the rewriting of BIOS starts, the status read/write section 10 sets the update flag 121 of the status register 12 to "1" to enter rewrite mode (step S3).

[0027]

10 The address conversion section 11 converts the address of the block to be rewritten by the value of the boot flag 120 of the status register 12. If the update flag 121 is "1" and the boot flag 120 is "0", then the address is converted into the address of the block 21, and BIOS data is rewritten. If the boot flag 120 is "1", then the address is converted
15 into [the address] of the block 20, and BIOS data is written (steps S5, S6).

[0028]

20 When the rewriting of the BIOS completes normally, the status read/write section 10 sets the update flag 121 to "0", and inverts the value of the boot flag 120 to "1" if currently "0", and to "0" if currently "1" (step S9), and the content of the status register 12 is written to the block 22 (step S10).

[0029]

25 By this, at the next startup the system can be started up from the block for which the BIOS data is rewritten.

[0030]

When the rewriting of the BIOS does not end normally, a reset is executed (step S8), but the boot flag 120 is not rewritten, so the system is started up using the BIOS data of the block for which the system has been started up with so far, which can prevent the status where the system cannot be started up even if the rewriting of the BIOS failed.

[0031]

Other embodiments of the present invention will now be described in detail. Fig. 4 is a block diagram depicting an example of the configuration of the address/boot conversion control section 1 according to the second embodiment of the present invention. Referring to Fig. 4, the detailed configuration of the address/boot conversion control section 1 in an embodiment of the present invention will be described.

[0032]

Referring to Fig. 4, according to the second embodiment of the present invention, a flag switch 3, which can directly switch the value of the boot flag 120 of the status register 12 immediately after power is turned on, is connected to the address/boot conversion control section 1 of the above mentioned embodiment shown in Fig. 2.

[0033]

By the flag switch 3, the boot flag 120 is set to a fixed value, "0" or "1". Then the address conversion section 11 reads the BIOS data in the block 20 or 21, depending on the value of the boot flag 120, and starts up the system.

[0034]

According to the second embodiment of the present invention, in a system where the hardware is the same but the settings of the CPU and each I/O are different, the BIOS is stored in one flash ROM, and is simply switched depending on the purpose of use. Therefore new hardware, where only the BIOS data is different, is unnecessary.

[0035]

According to the third embodiment of the present invention, the number of blocks in the flash ROM 2 is increased to N ($N > 4$), where $N-1$ number of BIOS data is stored, and the boot flags of the status register are increased accordingly. In this embodiment, the system can be started up with the $N-1$ number of BIOS which have different settings.

[0036]

[Effects of the Invention]

As described above, according to the present invention, the status where the system cannot be started up at the next system startup can be prevented with certainty even if the rewriting of the BIOS failed.

[0037]

This is because in the present invention, the flash ROM is comprised of a plurality of blocks where the BIOS data is stored, and BIOS is rewritten to a block which is different from the block from which the system is started up.

[Brief Description of the Drawings]

Fig. 1 is a diagram depicting the configuration of an embodiment of the present invention;

Fig. 2 is a diagram depicting the configuration of the address/boot conversion control section according to an embodiment of the present invention;

Fig. 3 is a flow chart depicting the processing flow of an embodiment of the present invention; and

Fig. 4 is a diagram depicting the address/boot conversion control section 1 according to an embodiment of the present invention.

[Explanation of Reference Numerals]

- 10 1 address address/boot conversion control section
- 2 flash ROM
- 3 flag switch
- 10 status read/write section
- 11 address conversion section
- 15 12 status register
- 120 boot flag
- 121 update flag
- 20, 21, 22 block

(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号

特開平11-306007

(43) 公開日 平成11年(1999)11月5日

(51) Int.Cl.⁶

G 0 6 F 9/06

13/10

識別記号

5 4 0

3 2 0

F I

G 0 6 F 9/06

13/10

5 4 0 C

5 4 0 M

3 2 0 A

審査請求 有 請求項の数 7 F D (全 7 頁)

(21) 出願番号 特願平10-131122

(22) 出願日 平成10年(1998)4月24日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 森光 弘

東京都港区芝五丁目7番1号 日本電気株式会社内

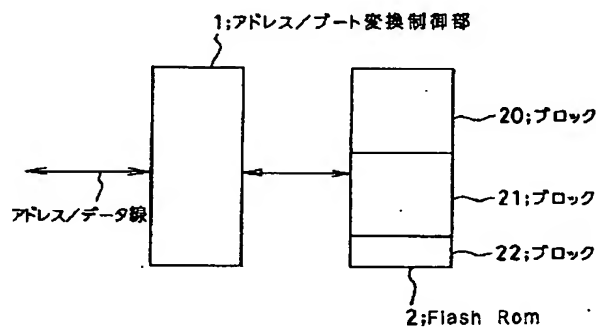
(74) 代理人 弁理士 加藤 朝道

(54) 【発明の名称】 B I O S書き換え方法及び方式

(57) 【要約】

【課題】 B I O Sデータの書換を失敗しても次回以降のシステムを立ち上げを可能とする B I O S書換方法及び方式の提供。

【解決手段】 フラッシュROMを B I O Sを格納するブロック 2 0、2 1 と、ステータス情報を格納するブロック 2 2 とに分け、システム立ち上げ時最初にブロック 2 2 から読出した情報に基づき B I O Sを読み出すブロックを選択してシステムを立ち上げ、 B I O S書換時には B I O Sを格納したブロック 2 0、2 1のうちシステム立ち上げに用いられたブロックとは別のブロックを新しい B I O Sで書き換え、新しい B I O Sで書換が正常に行われた場合にのみ、次回のシステム立ち上げ時、新しく B I O Sを書換えたブロックより B I O Sを読み出してシステムの立ち上げを行うように制御するアドレス／ポート変換制御手段 (1) を備える。



【特許請求の範囲】

【請求項1】書換可能な不揮発性半導体記憶装置を、それぞれBIOSを格納する複数の領域と、システム立ち上げに読み出す領域を決定する情報を含むステータス情報を格納し、システム立ち上げ時に最初に読み出される領域とに分け、

システム立ち上げ時、まず、前記ステータス情報を格納する領域から読み出した情報に基づきBIOSを読み出す領域を選択し該選択された領域からBIOSを読み出してシステムを立ち上げ、

BIOS書換時には、前記BIOSを格納した複数の領域のうち、システム立ち上げに用いられた前記領域とは別の領域を新しいBIOSで書き換え、

新しいBIOSで書き換えが正常に行われた場合にのみ、次のシステム立ち上げ時、前記新しくBIOSを書き換えた領域よりBIOSを読み出してシステムの立ち上げを行う、ことを特徴とするBIOS書き換え方法。

【請求項2】フラッシュROMを、システム立ち上げ時に最初に読み出されるステータス情報を格納するブロックと、BIOSを格納する複数のブロックと、に分け、システム立ち上げ時、ステータス情報を格納するブロックから読み込んだ情報からBIOSを読み出すブロックを選択し該選択されたブロックからBIOSを読み出してシステムを立ち上げ、

BIOS書換時には、システム立ち上げに用いられた前記ブロック（「元のブロック」という）とは別のブロックを新しいBIOSで書き換え、

前記新しいBIOSの書き換えが正常に行われた場合には、前記ステータス情報を格納するブロックの内容を更新し、

次回システム立ち上げ時、前記BIOSの書き換えが正常に行われた場合、前記新しくBIOSを書き換えたブロックよりBIOSを読み出してシステムの立ち上げを行い、一方、前記BIOSの書き換えが失敗した場合には、前記元のブロックよりBIOSを読み出してシステムの立ち上げを行う、ことを特徴とするBIOS書き換え方法。

【請求項3】書換可能な不揮発性半導体記憶装置を、BIOSを格納した第1、第2領域と、システム立ち上げ時に最初に読み出されるステータス情報を格納する第3領域と、に分割し、

システム立ち上げ時、最初に前記不揮発性半導体記憶装置内の前記第3領域に格納されたステータス情報を読み出してアドレスを制御し、前記第1又は第2領域のいずれかの領域に書き込まれているBIOSデータを読み出してシステムを立ち上げると共に、BIOSの書き換え時、前記不揮発性半導体記憶装置内の前記第1、第2の領域のうち、システムの立ち上げに使われた領域とは異なる領域に対して新しいBIOSの書き込みが行われる

ように制御し、さらに、前記第3領域を、前記第1、第2の領域のいずれからシステムを立ち上げるかに関する最新のステータス情報で更新するアドレス／ブート変換制御手段を備えた、ことを特徴とするBIOS書き換え方式。

【請求項4】前記アドレス／ブート変換制御手段が、前記第1、第2の領域のうちいずれの領域に格納されたBIOSを読み出してシステムの立ち上げを行うかを示すブートフラグと、BIOS書換モードを示すアップデートフラグとを含むステータス情報記憶手段と、

システムを立ち上げるに際して、まず最初に、前記不揮発性半導体記憶装置内の前記第3の領域からステータス情報を読み出してその内容を前記ステータス情報記憶手段に保存し、且つ、BIOS書き換え終了後に、前記ブートフラグの値を反転しステータス情報記憶手段の値を前記第3の領域に新しいステータス情報として書き込む手段と、

前記アップデートフラグが書換モードでない場合には、前記ブートフラグの値に応じて、前記第1又は第2の領域のいずれかにアクセスするようにアドレスを変換し、前記アップデートフラグの値が書換モードを示す場合には、前記第1、及び第2の領域のうちシステム立ち上げに際してBIOS読み出しに用いられた領域とは逆の領域にアクセスするようにアドレスを変換する手段と、を備えたことを特徴とする請求項3記載のBIOS書換方式。

【請求項5】前記ブートフラグの値を固定値に切り替え設定するための手段を、前記アドレス／ブート変換制御手段の外部に備えたことを特徴とする請求項3記載のBIOS書換方式。

【請求項6】書換可能な不揮発性半導体記憶装置を、BIOSを格納するためのN個（ただしNは3以上の整数）の領域と、システム立ち上げ時最初に読み出される前記ステータス情報を格納するためのN+1の領域に分割し、

前記不揮発性半導体記憶装置内の第N+1領域に格納されたステータス情報から、アドレスを制御して、前記第1乃至第N領域のいずれかの領域に書き込まれているBIOSデータを読み出してシステムを立ち上げ、

またBIOSの書き換え時、前記不揮発性半導体メモリ装置内の第1乃至第Nの領域のうち、システムの立ち上げに用いられた領域とは異なる領域に対して新しいBIOSの書き込みが行われるように制御し、さらに、前記第N+1領域を、第1乃至Nの領域のいずれからシステムを立ち上げるかに関する最新のステータスに更新するアドレス／ブート変換制御手段を備えた、ことを特徴とするBIOS書換方式。

【請求項7】前記書換可能な不揮発性半導体記憶装置が、ブロック毎に電気的に消去・書込みが可能なフラッシュROMよりなる、ことを特徴とする請求項3乃至5

のいずれかーに記載のBIOS書換方式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、パーソナルコンピュータ等情報処理装置のBIOS(basic input/output system)書き換方式に関し、特に、ブロック単位の消去・書き込み可能なフラッシュROMにBIOSを格納したシステムに用いて好適なBIOS書き換え方式に関する。

【0002】

【従来の技術】パーソナルコンピュータ等情報処理装置のハードウェア機能を制御する基本ソフトウェアシステムであるBIOSがマスクROM等に格納されている場合、その更新はマスクROMの交換を行うことで行われるが、近時、ブロック単位の書き換え可能なフラッシュROMにBIOSを格納して、BIOSの修正、機能追加等のアップデートを可能とした情報処理装置が製品化されている。この場合、BIOSを書き換えるためには、ユーザ自信で更新するか、もしくは保守契約をしている場合には保守担当者によって行われている。

【0003】

【発明が解決しようとする課題】しかしながら、BIOSの書き換えを行うには、BIOSを入手し所定の書き換えツールを起動して所定の手順に従って行う必要があり、ある程度の専門知識がないと、書き換えに失敗してしまう可能性がある。

【0004】また、書き換え途中に誤って電源を落としたり、フラッシュROM内のBIOSデータを書き換えツールによって消去した後にフラッシュROMにBIOSの書き込みが正常に行われなかった場合には、次回システムを正常に立ち上げることができない。これは、システム立ち上げ時、まず始めに、フラッシュROM内にあるBIOSデータを読み出して立ち上を行うが、このBIOSデータが無くなっているかもしくは壊れている場合、システム立ち上げが行えないためである。

【0005】なお、BIOS書き換え方式として、例えば特開平8-69376号公報には、BIOS ROM内容の書き換え途中で電源が断たれても再び動作させることができなくなってしまうことを回避するためのBIOS書き換え装置として、BIOSの格納する不揮発性メモリ素子の同一容量の上位・下位部分のいずれかを対応させるかをジャンプスイッチを用いて切り替えるアドレス切り替え回路を備え、BIOSは基本部分と拡張部分に分割し、基本部分の書き換えは、新しい基本部分を古い拡張部分を格納している部分に書き込み、書き込み完了後に、不揮発性メモリ素子の上位・下位部分を回路的に交換し、その後古いBIOS基本部分を格納したブロックを消去し新しい拡張部分を書き込むようにした方式が提案されている。後の説明で明らかとされるように、

本発明は、上記例えば特開平8-69376号公報記載の書き換え方式とは相違した方式を提供するものである。

【0006】したがって、本発明は、上記問題点に鑑みてなされたものであって、その目的は、たとえBIOSデータの書き換えに失敗しても次回以降のシステムを立ち上げを可能とするBIOS書き換え方法及び方式を提供することにある。

【0007】

10 【課題を解決するための手段】前記目的を達成するため、本発明に係るBIOS書き換え方法は、書換可能な不揮発性半導体記憶装置を、それぞれBIOSを格納する複数の領域と、システム立ち上げに読み出す領域を決定する情報を含むステータス情報を格納する領域とに分け、システム立ち上げ時、まず前記ステータス情報を格納する領域から読み出した情報に基づきBIOSを読み出すブロックを選択し該選択されたブロックからBIOSを読み出してシステムを立ち上げ、BIOS書換時には、BIOSを格納した複数の領域のうち、システム立ち上げに用いられた前記領域とは別の領域を新しいBIOSで書き換え、新しいBIOSで書き換えが正常に行われた場合にのみ、次回のシステム立ち上げ時、前記新しくBIOSを書き換えた領域よりBIOSを読み出してシステムの立ち上げを行うようにしたものである。

20 【0008】また本発明に係るBIOS書き換え方式は、書換可能な不揮発性半導体記憶装置を、BIOSを格納した第1、第2領域と、システム立ち上げ時に最初に読み出されるステータス情報を格納する第3領域と、に分割し、システム立ち上げ時、最初に前記不揮発性半導体記憶装置内の第3領域に格納されたステータス情報を読み出して、アドレスを制御し前記第1又は第2領域のいずれかの領域に書き込まれているBIOSデータを読み出してシステムを立ち上げると共に、BIOSの書き換え時、前記不揮発性半導体記憶装置内の第1、第2の領域のうち、システムの立ち上げに使われた領域とは異なる領域に対して新しいBIOSの書き込みが行われるように制御し、さらに、前記第3領域を、第1、第2の領域のいずれからシステムを立ち上げるかに関する最新のステータス情報で更新するアドレス/ブート変換制御手段を備えて構成される。

【0009】

30 【発明の実施の形態】本発明の実施の形態について以下に説明する。本発明は、フラッシュROM内を、システムを立ち上げるためのブロックと、BIOS書き換え用のブロックと、に分け、書き換えが正常に行われた場合にのみ、新しくBIOSを書き換えたブロックよりシステムを立ち上げるように制御するものである。

40 【0010】より詳細には、本発明のBIOS書き換え方法は、その好ましい実施の形態において、BIOSを格納するフラッシュROMを、システム立ち上げ時に最初

に読み出されるステータス情報を格納するブロックと、BIOSを格納する複数のブロックと、に分け、システム立ち上げ時、ステータス情報を格納するブロックから読み込んだ情報からBIOSを読み出すブロックを選択し該選択されたブロックからBIOSを読み出してシステムを立ち上げ、BIOS書き換え時には、システム立ち上げに用いられた前記ブロック（「元のブロック」という）とは別のブロックを新しいBIOSで書き換え、新しいBIOSの書き換えが正常に行われた場合には、ステータス情報を格納するブロックの内容を更新し、次回システム立ち上げ時、BIOSの書き換えが正常に行われた場合、新しくBIOSを書き換えたブロックよりBIOSを読み出してシステムの立ち上げを行い、一方、BIOSの書き換えが失敗した場合には、元のブロックよりBIOSを読み出してシステムの立ち上げを行うよう制御するものである。

【0011】本発明のBIOS書き換え方式は、その好ましい実施の形態において、図1を参照すると、フラッシュROM（2）を、BIOSを格納する第1、第2ブロック（20、21）と、システム立ち上げ時に最初に読み出されるステータス情報を格納する第3ブロック（22）と、に分割し、システム立ち上げ時に際して、まず、第3ブロックに格納されたステータス情報を読み出し、アドレスを制御して第1又は第2ブロックのいずれかの領域に書き込まれているBIOSデータを読み出すことでシステムを立ち上げ、またBIOSの書き換え時には、フラッシュROMの第1、第2ブロックのうち、システムの立ち上げに使われた領域とは異なる領域に対して新しいBIOSの書き込みが行われる（例えば第1ブロックでシステムを立ち上げた場合には第2ブロックにBIOSの書き換えを行う）ように制御し、さらに、第3ブロックを、第1、第2のブロックのいずれからシステムを立ち上げるかに関する最新のステータス情報で更新するアドレス／ブート変換制御手段（1）を備える。

【0012】すなわちアドレス／ブート変換制御手段（1）は、フラッシュROM（2）内のブロック（22）に格納されたステータス情報からアドレスを制御して、ブロック（20）又はブロック（21）のどちらかのBIOSデータからシステムを立ち上げるように制御するための機能手段を備える。

【0013】またBIOSの書き換え時、アドレス／ブート変換制御部（1）は、システムの立ち上げに使われたブロックとは異なるブロックへ書き込みが行われるように制御し、さらにブロック（22）を、最新のステータス情報に変更するように制御するための機能手段を備える。

【0014】本発明の実施の形態においては、フラッシュROM内に複数のBIOSデータが格納されたブロックを備え、BIOS書き換え時には、システムの立ち上げが

行われたブロックと異なるブロックに対してBIOSの書き換えを行うようにし、システムの立ち上げに用いられたブロックについては消去・書き込みが行われないため、たとえBIOSの書き換えを失敗しても次のシステム立ち上げ時には、システムが立ち上がらないという事態の発生を確実に回避することができる。以下実施例に即して詳説する。

【0015】

【実施例】本発明の実施例について図面を参照して以下に説明する。図1は、本発明の一実施例の構成を示す図である。

【0016】図1を参照すると、本発明の一実施例は、BIOSデータが書き込まれており、電源断時にも記憶内容が保持されるフラッシュROM2（一括消去型の電気的に消去及び書き込み可能な読み出し専用メモリ）に接続しているアドレス、データおよび制御線の間に、アドレス及びフラッシュROM2内の読み出すブロックを変換するアドレス／ブート変換制御部1を備えている。

【0017】フラッシュROM2は、BIOSデータが書き込まれ、読み出し・書き込み可能な2つのブロック20、21と、フラッシュROM2のステータス情報が格納されており、システムの立ち上げ時に最初に読み出されるブロック22と、を備えて構成されている。

【0018】図2は、本発明の一実施例におけるアドレス／ブート変換制御部1の構成の一例を示すブロック図である。図2を参照して、本発明の一実施例におけるアドレス／ブート変換制御部1の詳細な構成について説明する。

【0019】アドレス、データおよび制御線に接続してフラッシュROM2のブロック22からステータス情報のリード（読み出し）／ライト（書き込み）を制御するステータス読み書き部10と、ステータス読み書き部10によりリードされた値を格納できるリード／ライト可能なステータス（Status）レジスタ12と、ステータスレジスタ12をリードして判定し、アドレスを変換するアドレス変換部11と、を備えて構成されている。

【0020】ステータス読み書き部10は、ブロック20又はブロック21のBIOSデータからシステムを立ち上げるに際して、まず最初に、ブロック22からステータス情報をリードしてステータスレジスタ12に保存し、また書き換え終了後にステータスレジスタ12の設定値をブロック22に新しいステータス情報として書き込む機能を有する。

【0021】ステータスレジスタ12は、ブロック20のBIOSデータでシステムが立ち上がっている場合に「0」、ブロック21のBIOSデータの場合に「1」が設定されるブート（Boot）フラグ120と、書き換えモードの場合に「1」、それ以外の場合は「0」が設定されるアップデート（Update）フラグ121とを有する。

【0022】また、アドレス変換部11は、(1)アップデートフラグ121の値が「0」であり、且つブートフラグ120の値が「0」の場合には、ブロック20から、(2)アップデートフラグ121の値が「0」であり、且つブートフラグ120の値が「1」の場合には、ブロック21へアクセスするようにアドレスを変換し、アップデートフラグ121の値が「1」の場合には、ブート時に用いられたブロックとは逆のブロックにアクセスするようにアドレスを変換する機能を有する。

【0023】図3は、本発明の一実施例の動作の説明するためのフローチャートである。図1乃至図3を参照して、本発明の一実施例の動作について説明する。

【0024】まず、システムの電源を投入すると(ステップS1)、ステータス読み書き部10がブロック22からステータス(Status)情報を読み出し該情報をステータス(Status)レジスタ12に情報を格納する(ステップS2)。アドレス変換部11は、ブートフラグ120の値が「0」ならブロック20、「1」ならブロック21を読み出してシステムを立ち上げる。

【0025】その後、新しいBIOSのデータおよび書き換えツールの格納されたブートFD(フロッピーディスク)やブートディスクのようなブート媒体から立ち上がり書き換えが始まる。

【0026】BIOSの書き換えが始まると、書き換えモードにするため、ステータス読み書き部10が、ステータスレジスタ12のアップデートフラグ121を「1」にする(ステップS3)。

【0027】アドレス変換部11は、ステータスレジスタ12のブート(Boot)フラグ120の値により、書き換えるブロックのアドレスを変換する。アップデートフラグ121が「1」であり、ブートフラグ120が「0」なら、ブロック21のアドレスに変換して、BIOSデータを書き換える。また、ブートフラグ120が「1」ならブロック20のアドレスに変換してBIOSデータを書き換える(ステップS5、S6)。

【0028】BIOSの書き換えが正常に終了したら、ステータス読み書き部10が、アップデートフラグ121を「0」とし、ブートフラグ120の値が、それまで「0」なら「1」、「1」なら「0」に反転させ(ステップS9)、ブロック22に、ステータスレジスタ12の内容を書き込む(ステップS10)。

【0029】これにより、次回立ち上げ時は、新しくBIOSデータを書き換えたブロックから立ち上げることができる。

【0030】また、BIOSの書き換えが正常終了しなかった場合には、リセットが行われるが(ステップS8)、ブートフラグ120が書き換えられていないので、今までシステムが立ち上がったブロックのBIOSデータを用いてシステムの立ち上げが行われ、このため、BIOSの書き換えが失敗した場合でもシステム

を立ち上げることができないといった事態を回避することができる。

【0031】本発明の他の実施例について図面を参照して詳細に説明する。図4は、本発明の第二の実施例のアドレス/ブート変換制御部1の構成の一例を示すブロック図である。図4を参照して、本発明の一実施例におけるアドレス/ブート変換制御部1の詳細な構成について説明する。

【0032】図4を参照すると、本発明の第二の実施例においては、図2に示した前記実施例のアドレス/ブート変換制御部1に、ステータスレジスタ12のブートフラグ120の値を、電源投入後、すぐに、直接切り替えることのできるフラグ切り替え器3が接続されている。

【0033】フラグ切り替え器3により、ブートフラグ120は、「0」もしくは「1」の固定値に設定される。その後、アドレス変換部11は、ブートフラグ120の値によって、ブロック20もしくは21のどちらかのBIOSデータを読み出してシステムを立ち上げる。

【0034】本発明の第二の実施例は、ハードウェアは同じであるがCPUや各I/Oの設定が異なるシステムにおいて、BIOSを1つのフラッシュROM内に格納しておき、使用目的に応じて簡単に切り替えることができる。このため、BIOSデータだけが異なる新たなハードウェアを不要としている。

【0035】また、本発明の第三の実施例として、フラッシュROM2内のブロックをN個($N > 4$)に増やしてN-1個のBIOSデータを格納し、それに対応してステータスレジスタのブートフラグも増やすような構成としてもよい。この実施例は、N-1個の設定の違ったBIOSでシステムを立ち上げることができる。

【0036】

【発明の効果】以上説明したように、本発明によれば、たとえBIOSの書き換えを失敗しても次のシステム立ち上げ時、システムが立ち上がらないということを確実に回避することができる、という効果を奏する。

【0037】その理由は、本発明においては、フラッシュROM内に複数のBIOSデータが格納されたブロックを備え、システムの立ち上げが行われたブロックと異なるブロックに対して、BIOSの書き換えを行うようにしたためである。

【図面の簡単な説明】

【図1】本発明の一実施例の構成を示す図である。

【図2】本発明の一実施例におけるアドレス/ブート変換制御部の構成を示す図である。

【図3】本発明の一実施例の処理フローを説明するためのフローチャートである。

【図4】本発明の一実施例におけるアドレス/ブート変換制御部1の構成を示す図である。

【符号の説明】

1 アドレス/ブート変換制御部

2 フラッシュROM

3 フラグ切り替え器

10 ステータス (Status) 読み書き部

11 アドレス変換部

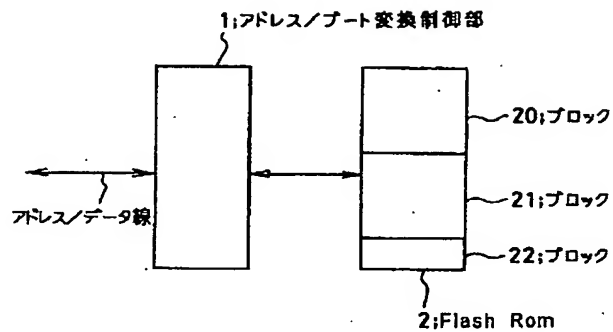
12 ステータス (Status) レジスタ

120 ブート (Boot) フラグ

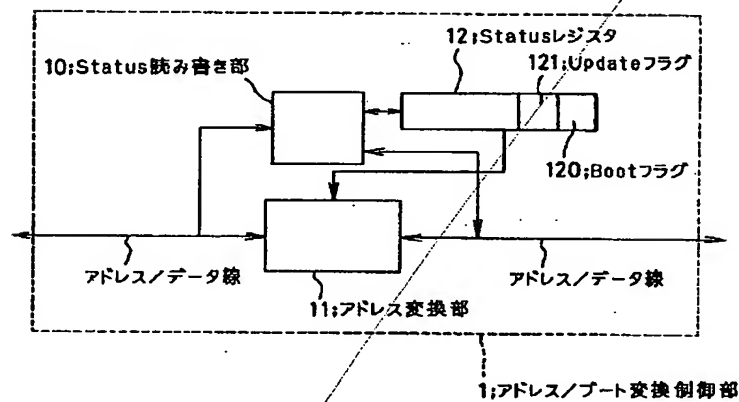
121 アップデート (Update) フラグ

20、21、22 ブロック

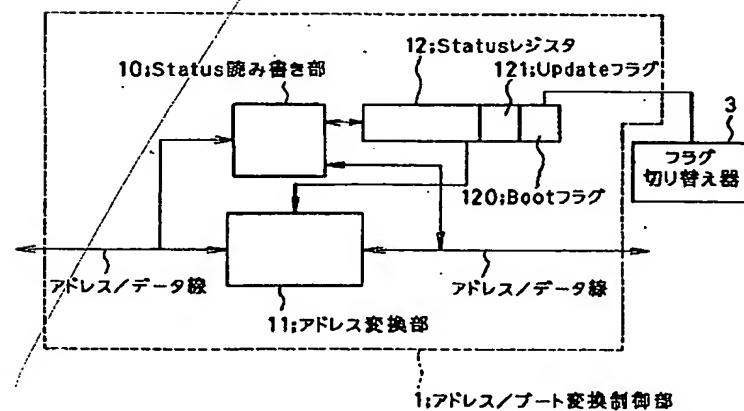
【図1】



【図2】



【図4】



【図3】

